



# Roy and Trending Topics

**Link submit:** <https://www.hackerearth.com/practice/data-structures/trees/heapspriority-queues/practice-problems/algorithm/roy-and-trending-topics-1/description/>

**Solution:**

C++	<a href="http://ideone.com/tGhn3A">http://ideone.com/tGhn3A</a>
Java	<a href="http://ideone.com/OP3oHI">http://ideone.com/OP3oHI</a>
Python	<a href="http://ideone.com/4DrrQc">http://ideone.com/4DrrQc</a>

**Tóm tắt đề:**

Cho bạn  $N$  topic với các ID và điểm phổ biến z-score. Mỗi ngày z-score sẽ thay đổi phụ thuộc vào việc tương tác của người dùng và điểm mới được tính như sau:

- Nếu là "Post" thì z-score +50 điểm.
- Nếu là "Like" thì z-score +5 điểm.
- Nếu là "Comment" thì z-score +10 điểm.
- Nếu là "Share" thì z-score +20 điểm.

Cho bạn danh sách những topic và các tương tác đối của topic đó. Hãy tính toán và in ra 5 topic có điểm chênh lệch cao nhất, điểm chênh lệch bằng điểm mới trừ điểm cũ. Nếu hai topic điểm bằng nhau thì bạn sẽ in ra topic nào có ID lớn hơn.

**Input:**

Dòng đầu tiên chứa số  $N$  ( $1 \leq N \leq 10^6$ ) – số lượng topic cần xử lý.

$N$  dòng sau, mỗi dòng là thông tin của một topic bao gồm các thông tin:

- Topic ID
- Điểm hiện tại z-score  $Z$
- Lượt post  $P$
- Lượt like  $L$
- Lượt comment  $C$
- Lượt share  $S$
- Ràng buộc:  $1 \leq ID \leq 10^9, 0 \leq Z, P, L, C, S \leq 10^9$ .

**Output:**

In ra 5 topic có điểm chênh lệch lớn nhất và điểm mới của topic đó.

**Ví dụ:**

8	1003 200
1003 100 4 0 0 0	1002 300
1002 200 6 0 0 0	1001 400
1001 300 8 0 0 0	999 200
1004 100 3 0 0 0	1007 150
1005 200 3 0 0 0	
1006 300 5 0 0 0	
1007 100 3 0 0 0	
999 100 4 0 0 0	

**Giải thích ví dụ:**

Bảng điểm sau khi đã tính toán lại của các topic như sau:

Topic ID	Old z-score	New z-score	Change in z-score
1003	100	200	200-100 = 100
1002	200	300	300-200 = 100
1001	300	400	400-300 = 100
1004	100	150	150-100 = 50
1005	200	150	150-200 = -50
1006	300	250	250-300 = -50
1007	100	150	150-100 = 50
999	100	200	200-100 = 100

Như vậy ta thấy những topic có điểm chênh lệch lớn hơn sẽ được in ra trước, nếu bằng nhau thì ID lớn hơn sẽ in ra trước.

**Hướng dẫn giải:**

Với mỗi topic bạn sẽ tính toán điểm chênh lệch = new z-score – old z-score. Sau đó lưu vào một cấu trúc topic.

Viết thêm một hàm so sánh để phù hợp với yêu cầu đề bài. Sau khi bỏ tất cả topic vào hàng đợi ưu tiên thì sẽ lấy 5 giá trị đầu in ra là kết quả của bài toán.

**Độ phức tạp:**  $O(N\log N)$  với N là số lượng topic cần xử lý.